$\TeX$, $\LaTeX$, and Farsi$\TeX$

Hamed Hatami
hatami@ce.sharif.edu

Behdad Esfahbod
behdad@sharif.edu

Computer Engineering Department
Sharif University of Technology
Tehran, Iran

May 2, 2003

“Things have changed in the past dacades.”, quotes Professor Donald Knuth from Bill Gates. The TEX system is known to have revolutionized the art of computer typesetting in the past three decades. After talking about the history of TEX, we will turn to LATEX and focus on what it offers an author, to reduce the time spent on style and work on the content.

What is TEX

- A computer language designed for use in typesetting; particularly, for typesetting math and other TEXnical (from greek *techne*, i.e. art/craft, the stem of *technology*) material.

- $\mathcal{T}\varepsilon\chi$, pronounced similar to "blecch", not to the state known for "Tex-Mex" chili.

History of TEX

- In the late 1970s, Donald Knuth of Stanford University, was not satisfied when he saw first samples of the new computer typesetting for the second volume of his multivolume opus *The Art of Computer Programming*.

- He set out to learn what were the traditional rules for typesetting math.

- A year after Knuth was invited by the American Math Society (AMS). The topic he presented was his new work on TEX.

History of TEX (continued)

- TEX came along just before the beginnings of the personal computer. It was developed on one of the last of the *academic* mainframes.

- It was very quickly ported to some early HP workstations and, as they emerged, the new personal systems.

- After some highly useful enhancements in about 1990, Tex handles the composition of many different languages according to their own traditional rules, and is for this reason quite widely used in eastern Europe.

Why did $\TeX$ Attract Scientists

- It was intended to be used directly by authors (and their secretaries) who are the ones who really know what they are writing about.

- It came from an academic source, and was intended to be available for no monetary fee.

- It was available on just about any computer and operating system.

Why did TeX Attract Scientists (continued)

- Other systems available at the time for mathematical composition were:
  - proprietary,
  - very expensive,
  - often limited to specific hardware,
  - if WYSIWYG*, the same expression in two places in the same document might very well not look the same, never mind look the same if processed on two different systems.

*What You See Is What You Get

Main Features of TEX

- A TEX system can stand on its own, provided all the fonts one needs are available. It does not require any other application.

- The TEX macro language is an interpreted one:
  - It produces a *device independent* output file in `.dvi` format.
  - It can be run on any operating system.
  - Running TeX on different platforms result in exactly identical outputs.

- TEX and its friends are available freely with source code. Moreover, the TEX itself is not even copyrighted.

Main Features of $\TeX$ (continued)

- Instead of the user, $\TeX$ automatically tries thousands of combinations of line breaks and page breaks on every job, picking the most charming one.

- You do not need to specify the exact appearance of everything. Whenever you leave something unspecified, it knows how to conform to standard practice of the used stylesheet.

- There are many tools, fonts, and macros developed for $\TeX$. Moreover, there is a $\TeX$ engine named pdf$\TeX$, which produces `pdf` files directly.

Main Features of TeX (continued)

- The command language is very low level (skip so much space, change to font X, set this string of words in paragraph form, ...), but is amenable to being enhanced by defining macro commands to build a very high level user interface.

- TeX has a very powerful algorithmical base. Some of the algorithms in TeX, for example the paragraph breaking algorithm, have not been bettered in any of the composition tools devised in the years since TeX appeared.

- TeX has all the paraphernalia of academic publishing, e.g. footnotes, floating insertions (figures and tables), etc.

Main Features of $\TeX$ (continued)

- A text written in $\TeX$ is a plain ASCII file with markup codes inserted, for example

```
$$ % begin displayed math
\left(\prod^{n}_{i=1}\prod^{c_i-1}_{j=0}
(n-j)!^{\frac{1}{n-j}}\right)
\le
\frac{ n!^n e^{n(3+\frac{\ln(2\pi n)^2}{4})} }
{(n-\frac{k}{n})!^n e^k}
$$ % end displayed math
```

will produce:

$$
\left(\prod^{n}_{i=1}\prod^{c_i-1}_{j=0}
(n-j)!^{\frac{1}{n-j}}\right)
\le
\frac{ n!^n e^{n(3+\frac{\ln(2\pi n)^2}{4})} }
{(n-\frac{k}{n})!^n e^k}
$$

Main Features of TEX (continued)

- TEX files can be typed, viewed, and edited with any file editor or word processor.

- TEX You do not have to worry about whether subtle details of typography are visible on the screen. Instead, you type TEX commands and TEX will set exactly what you tell it to.

- A completely separate file called a style sheet (actually a set of TEX macros) defines what the commands like `\chapter` and `\section` and their kin, actually should do. The typist does not have to worry about the physical layout or consistency of chapter or section headings on the screen.

Main Features of TEX (continued)

- It is designed for archival purposes. The same document will produce exactly the same output ten years later, with the TEX of that era.
- It is designed for batch processing. The input file to TEX is in a text file, so you can easily create your project result tables automatically as a TEX input file, process it, and get a nice `pdf`.
- ...

TEX and Other Systems

- Alternatives to TEX are FrameMaker and Adobe InDesign, which both are professional typesetting systems.

- Microsoft Word is not counted as a professional typesetting system, but a word processor.

- TEX both produces the high quality of a typesetting system, and holds the ease of use of a word processor.

TeX versus Microsoft Word

- TeX's math mode is a thing of beauty. Equations come out looking perfect.

- Unlike Word files, TeX documents are small and clean.

- In *What You See Is What You Get* word processors, it is very hard to write a large consistent text.

- TeX versions are compatible. A given Word file produces different results on different versions of Word, so all the painstaking effort you put in getting a perfect document in version 7 is broken in version 8.

TeX versus Microsoft Word (continued)

- Unlike TeX, Microsoft Word is not available for all operating systems.
- There are no TeX *macro* viruses. You can safely receive TeX documents by email and not worry about it reading your address book and mailing copies of itself to all your friends.
- The typesetting algorithm employed by Microsoft Word sacrifices quality to the speed required for setting and resetting type of the user's input in real time. The final product is greatly inferior to that of a real typesetting system.

When Should We Use It

- TEX is the best tool for writing a scientific text, but

- If you want a tool for producing a newspaper or a novel or a slick advertisement or a letter to Aunt Henrietta, TEX is not the tool for you.*

- Professor Knuth in an interview has said that:

  > I never expected TeX to be the universal thing that people would turn to for the quick-and-dirty stuff. I always thought of it as something that you turned to if you cared enough to send the very best.

*TEX is actually in use for producing many scientific and TEXnical magazines and journals.

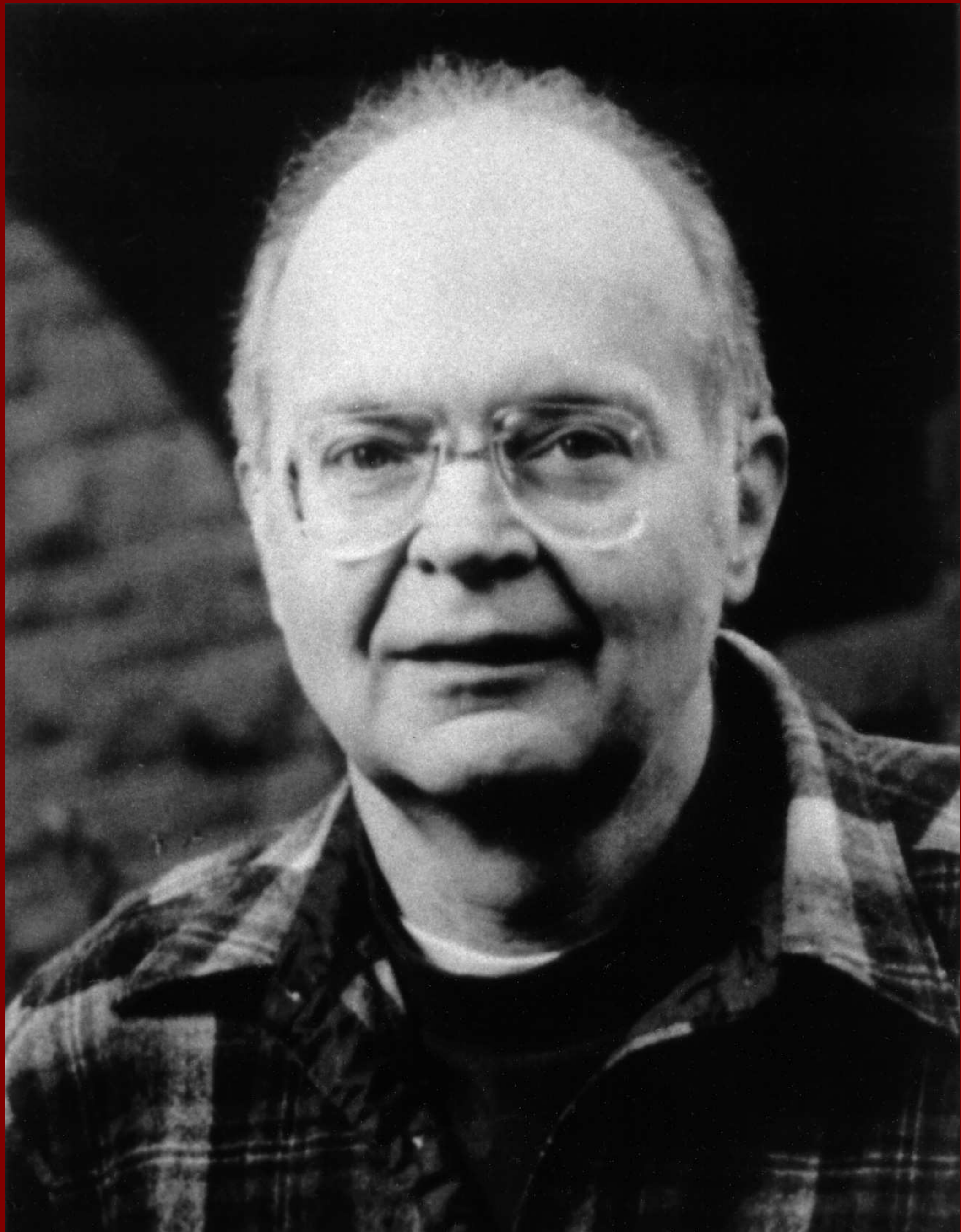
Bugs of $\TeX$

- Donald Knuth, a professor of computer science at Stanford University and the author of numerous books on computer science and the TeX composition system, rewards the first finder of each typo or computer program bug with a check based on the source and the age of the bug.

- Typos and other errors in books typically yield $2.56 each once a book is in print (pre-publication *bounty-hunter* photocopy editions are priced at $.25 per).

- Program bugs' cost rise by powers of 2 each year from $2.56 to a maximum of $327.68.

- So you can imagine how bug-free is $\TeX$!



Donald Ervin Knuth

What is LaTeX

- The TeX language is very low level.
- Originally TeX comes with a package called plain.
- There are many high level macro packages available.
- The most popular is LaTeX, another one is ConTeXt.
- LaTeX knows how to format each logical part of the document.
- LaTeX is Leslie Lamport's high level definition of a document, later developed by the LaTeX Project Team.

Structure of a $\LaTeX$ Document

- Starts with a preamble. The preamble is logically structured as below:
  - The document class definition (e.g. `\documentclass{article}`).
  - Include packages used (e.g. `\usepackage{color}`).
  - Configuration (e.g. `\textwidth=10in`).
  - User macro definitions (e.g. `\definecolor{backgroundcolor}{rgb}{.5,0,0}`).
- The document body. Starts with `\begin{document}` and ends with `\end{document}`.
- Everything after the body is ignored.

$\LaTeX$ Document Classes

- There are some predefined classes:
  - **article**
  - **report**
  - **book**
  - **letter**
  - **slides**
- It is always possible to define new classes based on availble ones, or from scratch.

Structure of the Document Body

Structure of the document body depends on the document class used, but for articles, reports, and books it is something like this:

- Title, Author, and Date definitions
- The title of the document
- The table of contents
- The abstract
- Parts, Chapters, Sections, Subsections, etc, in a nested manner
- Appendices
- The Bibliography
- The Index

Structure of a Simple Article

```
\documentclass{article} % the standard class ``article''
\begin{document}
\maketitle
\begin{abstract} ...  \endabstract
\section{...}
\section{...}
   \subsection{...}
      \subsubsection{...}
\section{...}
\begin{thebibliography} ...  \end{thebibliography}
\end{document}
```

Structure of a Simple Book

```
\documentclass{book} % the standard class ``book''
\begin{document}
%-------------------- front matter of the document
\maketitle
  \section*{...}     % e.g.  section named like ``Preface''
\tableofcontents     % chapter with the table of contents
\listoffigures       % chapter with the list of figures
\listoftables        % chapter with the list of tables
%-------------------- body of the document
\part{...}
\chapter{...}
   \section{...}
\chapter{...}
\part{...}
%-------------------- back matter of the document
\appendix            % following chapters are appendices
\chapter{...}
\chapter{...}
\begin{thebibliography} ...  \end{thebibliography}
\begin{theindex}        ...  \end{theindex}
\end{document}
```

Other Things TEX and LATEX Take Care of

- Footnotes* (e.g. `Footnotes\footnote{This is a footnote}`)
- Floats (Tables and Figures)
- Equations
- Theorems and friends
- References
- Index terms
- Citations

*This is a footnote

Properties Shared by All Objects

- New objects of all kinds can be defined.
- All objects are automatically numbered in its own way.
- Numbering of different objects can be defined to follow your own style.
- Objects without number or caption can be declared.

Float Objects

- A float object is a box with a type and caption.
- Table and Figure are already defined as floats.
- New floats can be defined. For example, there is a package to define an Algorithm float object.
- List of floats of a type can be placed anywhere in the document.
- Floats are automatically placed in top of a page, or at the end of the chapter or document.
- A float can be forced to be placed where it is defined.

A Simple Float

```
\newfloat{algorithm}{h}{alg}
\floatname{algorithm}{Algorithm}
\begin{algorithm}
\caption{No comments}
\label{errless}
\begin{center}
\begin{enumerate}
\item Err and err and err again
\item but less and less and less
\end{enumerate}
\end{center}
\end{algorithm}
```

1. Err and err and err again

2. but less and less and less

Algorithm 1: No comments

Theorem-like Objects

- These are objects like Theorem, Lemma, Corollary, Congecture, etc.

- New theorem-like objects can be defined.

- Like all other objects, they can be numbered globally, or inside a part, chapter, section, etc.

- They are defined simply by `\newtheorem{lemma}{Lemma}`

- They can be used by `\begin{lemma}\label{texlovers}Anyone ... \end{lemma}`:

  Lemma 1 *Anyone heard of TEX would use it to typeset her thesis.*

Refrences

- A point to be refrenced is defined by `\label{labelname}`.
- To refer to a defined lable, you use `\ref{labelname}`.
- LaTeX knows what kind of object the label is refering to and puts the object's number.
- Refrences to page numbers of objects is also supported. Moreover, LaTeX can also handle refrences to objects in the same page, facing page, next or previous page.
- For example `Lemma \ref{texlovers} and Algorithm \ref{errless}` will produce: Lemma 1 and Algorithm 1.

Indexes

- An occurence of an index term in the document is identified by `\index{term}`.

- A program called `makeindex` takes care of merging, sorting, and preparing the index.

- The `\begin{theindex} ... \end{theindex}` puts the index in the document.

Citations

- Citations in the text are made by `\cite{refname}`.
- References are then defined in the `\begin{thebibliography} ... \end{thebibliography}`.
- There is a program named BibTEX, which can extract used refrences from a big set of reference definition file.

The Slides Class

- Each slide is placed between `\begin{slide} ... \end{slide}`.

- The slides come out highly readable, and well spaced.

- As there is almost no fancy feature, the audience's focus will be on the content.

- There is a package* which enables parts of an slide to become visible in the next step.

*`stepslid`, by Behdad Esfahbod

What is FarsiTeX

- FarsiTeX is a Persian enabled version of LaTeX.
- The current release of FarsiTeX is based on the old LaTeX2.09. The current version of LaTeX is LaTeX 2_ε.
- FarsiTeX supports almost all features of article, book, report, and letter classes, but not slides.
- All you need to use FarsiTeX is the editor and macros. The rest is simply LaTeX.

Where to Find Them

- TeX Users Group's (TUG) Home Page at `http://www.tug.org/` is the best place to look for TeX related tools and articles.

- TeX, LaTeX, and all of their tools, fonts, etc can be found on Comprehensive TeX Archive Network (CTAN) set of CDs, downloadable from `http://dante.de/`.

- FarsiTeX, is available from `http://www.farsitex.org/`.

Where to Find Them (continued)

- There are many distributions:
  - TEXlive, by Sebastian Rahtz, is a set of CDs, ready to run under Windows, Linux, MacOS, etc. It can be downloaded from `http://dante.de/` too.
  - MiKTEX by Christian Schenk, is a distribution for Microsoft Windows. It is available from `http://www.miktex.org/`.
  - teTEX by Thomas Esser, is a distribution for Unices (based on Karl Berry's kpathsea, and web2c), but ported to Windows under Cygwin, and also as another distribution named fpTEX, by Fabrice Popineau. It is distributed by almost all Linux distributions, as well as Cygwin.

Conclusions

The LaTeX system has become the *de-facto* standard for typesetting mathematical and technical papers, reports, and books. We have shown some of the aspects of TeX, and LaTeX and mostly attracts authors all around the world to set type of their work themselves with LaTeX. Interested audience are encouraged to read *LaTeX, A Document Preparation System*, and then get their hands on LaTeX, to feel what is known as the spirit of TeX.

FarsiTeX, as a Persian enabled LaTeX system, is widely used in Iranian academic centers, but as we mentioned, it is far from a modern Persian LaTeX system. Work is getting done to port FarsiTeX to the latest standards of LaTeX system.

Refrences

Knuth, Donald. *The TEXbook*. Addison-Wesley, 1996.

Knuth, Donald. *Digital Typography*. CLSI Publications, 1999.

Lamport, Leslie. *LATEX, A Document Preparation System*. Addison-Wesley, 1994.

Goossens, Michel, Frank Mittelback, and Alexander Samarin. *The LATEX Companion*. Addison-Wesley, 1994.

TEX Users Group Home Page. Available at `http://www.tug.org/`.

Nordhaug, Hans Fredrik. "Why TEX". Available at `http://www.mi.uib.no/~hansfn/tex/advocacydft.pdf`.